REMARKS

Claims 1-14 and 30-39 are pending and claims 15-29 are withdrawn. Applicants have amended claims 1, 11-12, 14, 30-31, 33 and 39 to more clearly define the present invention without changing the scope of the invention.

Claims 1-3, 5-7, 9, 11-12, 14, 30-35 and 38-39 are rejected under 35 U.S.C. 103(a). In particular, the Examiner stated:

> Claims 1-3, 5-7, 9, 11-12, 14, 30-35 and 38-39 are rejected under 35 U.S.C. 103(a) as being unpatentable over Guthridge et al. (US 7,124,131 B2), granted October 17,2006, filed April 29, 2003, hereinafter Guthridge, in view of Taylor (US 7,107,267 B2) Method, System, Program, and Data Structure for Implementing a Locking Mechanism for a Shared Resource, granted September 12, 2006, filed January 31,2002, hereinafter Taylor.
>
> In regard to claim 1, Guthridge discloses **a method for attempting to access a first data entity in a file system** as a method for reasserting a lock in a distributed file system (column 2, lines 1-2), the method being performed by a first computing entity, the file system also including one or more additional data entities that are concurrently accessible to at least one other computing entity, the file system including an owner field that can be used to determine whether the first data entity is leased by a computing entity and a time field that can be used to determine whether a lease for the first data entity has expired, the method comprising:
>
> **the first computing entity attempting to obtain a lease for itself on the first data entity by performing the reading the owner field** as a query is conducted to determine if a lock manager data structure exists (Guthridge, column 4, lines 29-31, Fig. 3A, element 70) and:
>
> **if the owner field indicates that the first data entity is not currently leased, the first computing entity writing to the owner field to indicate an assumption of a lease of the first data entity** as if the lock data structure does not exist, a new lock manager data structure for the identified object is created (Guthridge, column 4, lines 33-35, Fig. 3A, element 76), client node identifier associated with the lease (Guthridge, column 6, lines 57-58), and **writing to the time field to indicate when the lease expires** as return file system epoch number of the lock (Guthridge, column 4, lines 58-59); or
>
> **if the owner field indicates that the first data entity has been leased, the first computing entity reading the time field** as a response to query indicating a current lease, return epoch number of the lock (Guthridge, column 4, line 58-59, Fig. 3A, element 82) and:
>
> **if the time field indicates that the lease has expired, the first computing entity writing to the owner field to break the existing lease** as if a client node requests a lock with an expired lease, requesting client node may recover the lock lease (Guthridge, column 4, lines 26-29), expired lease is deleted (Guthridge, column 5, line 32-33) and **to indicate an assumption of a new lease** as flag is set (Guthridge, column 5, lines 31-32, Fig. 3B, element 96) and the **first computing entity writing to the time field to indicate when the new lease expires** as locks have a lease for a limited time period (Guthridge, Abstract, lines 2-3), hold lock for a set lease period (Guthridge, column 8, lines 23-24); or
>
> **if the time field indicates that the lease is still active, concluding that the first data entity is currently unavailable** as if there is a conflict, the lock request is denied (Guthridge, column 6, lines 66-67); and
>
> **if a lease is obtained, the first computing entity accessing the first data entity while the lease is in effect** as once the client node holds a distributed lock the client node can access the data (Guthridge, column 1, lines 30-32), however Guthridge does not specifically disclose reading and writing to a time field.

On the other hand, Taylor discloses a lease data structure with a lease start time (Taylor, column 5, line 46, Fig. 3, element 104), lease length (Taylor, column 5, lines 45-46, Fig. 3, element 106) and determining that the lease time has expired (Taylor, column 2, line 34).

It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to modify the teaching of Guthridge to include the lease time of Taylor for **reading the time field,** determining **if the time field indicates that the lease is still active** and **writing to the time field** because a granted access would cease when the lease expired (Taylor, column 5, lines 44-45). It is also noted that both Guthridge and Taylor are from file management, and more specifically access management (Guthridge, Abstract: lines 1-2, locks in a distributed file system, Taylor, Abstract: line 2-3, locking mechanism to control access to a shared resource).

In regard to claim 2, Guthridge discloses the method wherein the **first data entity is a file** as client can access the data for the file (column 1, lines 31-32).

In regard to claim 3, Guthridge discloses the method of wherein the **first data entity includes metadata** as metadata information for the requested file (column 1, line 26) and **the owner field is located in this metadata** as attributes of a file, such as owner, group, mode, etc. maintained in client node data cache (column 7, lines 17-18).

In regard to claim 5, Guthridge discloses the method of claim 1, wherein the step of the **first computing entity writing to the owner field to indicate an assumption of a lease of the first data entity comprises the first computing entity writing a data value to the owner field that uniquely identifies the first computing entity** as a unique client node identifier is assigned to the client node when the lease is established (Guthridge, column 3, lines 19-21).

In regard to claim 6, Guthridge discloses the method wherein the **data value that uniquely identifies the first computing entity is determined autonomously by the first computing entity** as manager adapted to control a lock version number (Guthridge, column 2, lines 16-18).

In regard to claim 7, Guthridge discloses the method wherein **the owner field indicates that the first data entity is not currently leased when the owner field contains a value of zero** as lock is downgraded to 'None' indicating the lock should be released completely (Guthridge, column 7, line 44-45).

In regard to claim 9, Guthridge discloses the method **wherein the first computing entity determines whether a prior lease has expired by reading a first value from the time field** as determine if the client requesting the lock already holds a lock (Guthridge, column 4, lines 48-49), query will return epoch number (Guthridge, column 4, lines 56-59), **delaying for a predetermined lease period** as new lock requests are denied during a lock reassertion grace period (Guthridge, column 4, lines 19-20) **and reading a second value from the time field** as current epoch number of file system is read (Guthridge, column 6, line 18, Fig. 4A, element 144), **wherein the first computing entity determines that the prior lease has expired if the second value is the same as the first value** as determine if epoch number of file system is equivalent to epoch number provided by client node (Guthridge, column 6, lines 26-28, Fig. 4A, element 150), and **the first computing entity determines that the prior lease has not expired if the second value is different from the first value** as negative response will result in denial of lock assertion (Guthridge, column 6, lines 31-32).

In regard to claim 11, Guthridge discloses the method wherein, **if the first**

computing entity concludes that the first data entity is currently unavailable as if the lease has not expired (Guthridge, column 5, line 29, lines 33-34), **the first computing entity further writes an entry to queue owner field in a *list* to indicate an interest in accessing the first data entity** as the client node requesting the list is added to a list (Guthridge, column 5, line 35), however Guthridge does not specifically disclose **a queue.** On the other hand, Taylor discloses a resource queue (Taylor, column 4, line 31, Fig. 3, element 50).

It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to modify the teaching of Guthridge to include the resource queue of Taylor for **the first computing entity further writes an entry to queue owner field in a queue to indicate an interest in accessing the first data entity** because a resource queue provides a list of I/O requests with respect to a shared resource (Taylor, column 4, lines 34-37).

In regard to claim 12, Guthridge does not specifically disclose the method **wherein first the computing entity also writes to a queue time field to indicate a period of time for which the entry to the queue owner field is valid.** On the other hand, Taylor discloses a lease length (column 5, line 57, Fig. 3, element 118).

It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to modify the teaching of Guthridge to include the lease length of Taylor for **writes to a queue time field to indicate a period of time for which the entry to the queue owner field is valid** because if a process granted access would cease access operations when the lease expires (Taylor, column 5, lines 44-45).

In regard to claim 14, Guthridge discloses the method wherein, **if a lease is obtained, the first computing entity also sets a renewal timer** as locks have a lease for a limited time period (Guthridge, Abstract, lines 2-3) **and, after the renewal timer expires, the first computing entity renews the lease by writing a new value to the time field** as client node may reassert a lock for a lease that has expired (Guthridge, column 6, lines 13-14, Fig. 4A, element 158).

In regard to claim 30, Guthridge discloses **a method for attempting to access a first data entity in a file system** as a method for reasserting a lock in a distributed file system (Guthridge, column 2, lines 1-2), the method being performed by a first computing entity, the file system also including one or more additional data entities that are concurrently accessible to at least one other computing entity, the file system including an owner field that can be used to determine whether the first data entity is in use by a computing entity, the method comprising:

the first computing entity reading the owner field and determining whether the first data entity is in use by a computing entity as a query is conducted to determine if a lock manager data structure exists (column 4, lines 29-31, Fig. 3A, element 70);

if the first data entity is not in use by a computing entity, the first computing entity writing to the owner field to take control of a lock on the first data entity as if the lock data structure does not exist, a new lock manager data structure for the identified object is created (column 4, lines 33-35, Fig. 3A, element 76), client node identifier associated with the lease (column 6, lines 57-58); and

if control of the lock is obtained, the first computing entity accessing the first data entity as once the client node holds a distributed lock the client node can access the data (column 1, lines 30-32); or

if control of the lock is not obtained as if the lease has not expired (Guthridge, column 5, line 29, lines 33-34), **the first computing entity writing an entry to a *list* owner field to indicate an interest in accessing the first data entity and waiting for an opportunity to access the first data entity** as the client node requesting the list is added to a

list (Guthridge, column 5, line 35), however Guthridge does not specifically disclose **a queue**. On the other hand, Taylor discloses a resource queue (Taylor, column 4, line 31, Fig. 3, element 50).

It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to modify the teaching of Guthridge to include the resource queue of Taylor for **writing an entry to a queue owner field to indicate an interest in accessing the first data entity and waiting for an opportunity to access the first data entity** because a resource queue provides a list of I/O requests with respect to a shared resource (Taylor, column 4, lines 34-37).

In regard to 31, Guthridge discloses the method further comprising, **if the first data entity is in use by a computing entity, the first computing entity reading a time field to determine whether a lease on the data entity has expired** as a response to query indicating a current lease, return epoch number of the lock (Guthridge, column 4, line 58-59, Fig. 3A, element 82) and, **if the lease has expired, the first computing entity writing to the owner field to break the existing lease** as if a client node requests a lock with an expired lease, requesting client node may recover the lock lease (Guthridge, column 4, lines 26-29), expired lease is deleted (Guthridge, column 5, line 32-33) **and to indicate an assumption of a new lease of the first data entity** as flag is set (Guthridge, column 5, lines 31-32, Fig. 3B, element 96).

In regard to claim 32, Guthridge discloses the method **wherein the first computing entity determines whether the lease has expired by reading a first value from the time field** as determine if the client requesting the lock already holds a lock (Guthridge, column 4, lines 48-49), query will return epoch number (Guthridge, column 4, lines 56-59), **delaying for a predetermined lease period** as new lock requests are denied during a lock reassertion grace period (Guthridge, column 4, lines 19-20) **and reading a second value from the time field** as current epoch number of file system is read (Guthridge, column 6, line 18, Fig. 4A, element 144), **wherein the first computing entity determines that the lease has expired if the second value is the same as the first value** as determine if epoch number of file system is equivalent to epoch number provided by client node (Guthridge, column 6, lines 26-28, Fig. 4A, element 150), and **the first computing entity determines that the lease has not expired if the second value is different from the first value** as negative response will result in denial of lock assertion (Guthridge, column 6, lines 31-32).

In regard to claim 33, Guthridge discloses the method further comprising, if the first data entity is not in use by a computing entity, in addition to writing to the owner field to take control of the lock on the first data entity, **the first computing entity writing to a field to indicate when a lease of the first data entity expires** as locks have a lease for a limited time period (Guthridge, Abstract, lines 2-3), hold lock for a set lease period (Guthridge, column 8, lines 23-24). However, Guthridge does not specifically disclose the **first computing entity writing to a time field.**

On the other hand, Taylor discloses setting a lease start time in a lease data structure (Taylor, column 2, lines 50-51, Fig. 3, element 104).

It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to modify the teaching of Guthridge to include the start lease time of Taylor because a granted access would cease when the lease expired (Taylor, column 5, lines 44-45).

In regard to claim 34, Guthridge discloses the method **wherein the first data entity is a file** as client can access the data for the file (Guthridge, column 1, lines 31-32).

In regard to claim 35, Guthridge discloses the method wherein **the first data entity includes metadata** as metadata information for the requested file (Guthridge, column 1, line 26) **and the owner field is located in this metadata** attributes of a file, such as owner, group, mode, etc. maintained in client node data cache (Guthridge, column 7, lines 17-18).

In regard to claim 38, Guthridge discloses the method wherein **the first computing entity autonomously determines a data value that uniquely identifies the first computing entity** as manager adapted to control a lock version number (Guthridge, column 2, lines 16-18) and **the first computing entity assumes a lock on the first data entity by writing the unique data value into the owner field** as a unique client node identifier is assigned to the client node when the lease is established (Guthridge, column 3, lines 19-21).

In regard to claim 39, Guthridge and Taylor disclose the method further comprising, if control of the lock is not obtained, in addition to writing an entry to a queue owner field to indicate an interest in accessing the first data entity, however Guthridge does not specifically disclose **the first computing entity writing to a time field to indicate a period of time for which the entry to the queue owner field is valid**. On the other hand, Taylor discloses a lease length (Taylor, column 5, lines 46-47, Fig. 3, element 106).

It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to further modify the teaching of Guthridge to include the lease length of Taylor for **writes to a queue time field to indicate a period of time for which the entry to the queue owner field is valid** because if a process granted access would cease operations when the lease expires (Taylor, column 5, lines 44-45).

Applicants have amended claims 1, 11-12, 14, 30-31, 33 and 39 to more clearly define the present invention without changing the scope of the invention. As such, Applicants respectfully traverse the Examiner's rejection.

Discussion of Aspects of the Invention

Applicants respectfully submit that paragraph [0032] of the specification begins a description of an embodiment of the present invention. "The system of figure 2 includes the servers 10, 12 . . . 18 and the data storage unit 40 of figures 1A and 1B [i.e., the prior art]. The data storage unit 40 includes a file system 42, which may generally be the same as the file system 41, except for the addition of certain specific locking information that is described below." Continuing in paragraph [0033], the specification states: "Each of the servers 10, 12 and 18 is connected to the data storage unit 40 in some manner. These connections are illustrated in figure 2 as general data links 36. ... The data links 36A, 36B and 36N may be part of a data storage network 34, which may be substantially the same as the data network 32 shown in figures 1A and 1B. For example, the data storage network 34 may comprise a conventional SAN." And, further continuing in paragraph [0035], the specification states: "The only aspects of the system of figure 2 that must be unique in order to implement this invention are some software routines on the servers 10, 12 and 18 for

implementing some methods described below and some data fields within the file system 42." And, still further continuing in paragraph [0036], the specification states: "Figure 2 also shows that the servers 10, 12 and 18 may each include one or more VMs 13. This invention may be particularly advantageous when implemented in computer systems in which multiple VMs, such as the VMs 13, execute on multiple servers, such as the servers 10, 12 and 18, and share a common file system, such as the file system 42."

The data storage unit is described as follows in paragraph [0037] of the specification: "Turning now to figure 3, the data storage unit 40 is shown, including the file system 42A. The file system 42A may comprise a conventional file system, including a plurality of files of various types, typically organized into one or more directories. The file system 42A may include metadata that specifies information about the file system, such as some data structure that indicates which data blocks in the file system remain available for use, along with other metadata indicating the directories and files in the file system, along with their location. Each file and directory typically also has metadata associated therewith, specifying things such as the data blocks that constitute the file or directory, the date of creation of the file or directory, etc." **Importantly**, as set forth in paragraph [0038]: "To implement the invention, various additional data fields of metadata are preferably added to the file system 42A, such as a plurality of fields that are shown in figure 3. First, the file system 42A includes a file system lock 44, which includes an owner data field 45A and a time data field 45B. The owner data field 45A may be a unit of data, such as a byte, a word, etc., that is used to identify a computing entity that owns or possesses the lock 44. Possessing the lock 44 gives a computing entity exclusive access to the configuration data of the file system 42A. The owner data field 45A may contain a zero or some other special value to indicate that no computing entity currently owns the lock, or it may contain an identification (ID) value of one of the computing entities to indicate that the respective computing entity currently owns the lock. For example, each of the servers 10, 12 and 18 may be assigned a unique ID value, which could be inserted into the owner field 45A to indicate that the respective server owns the lock 44 for the file system 42A. A unique ID value need not be assigned manually by a system administrator, or in some other centralized manner. Instead the ID values may be determined for each of the servers 10, 12 and 18 in a simpler, more automated manner, such as by using the server's IP address or the MAC (Media Access Control) address of the server's network interface card, by using the World Wide Name

(WWN) of the server's first HBA or by using a Universally Unique Identifier (UUID). For the rest of this description, it will be assumed that a zero is used to indicate that a lock is not currently owned, although other values may also be used for this purpose."

Discussion of Guthridge et al.

Guthridge et al. discloses at col. 1, lines 13-33: "FIG. 1 is a prior art diagram 10 illustrating a SAN 5 with multiple server nodes 12 and 14, and multiple client nodes 16, 18, and 20. The SAN has metadata space 25 reserved for the server nodes 12 and 14. Each of the client nodes may access an object or multiple objects stored on the file data space 27 of the SAN, but may not access the metadata space. In opening the contents of an existing file object on the storage media in the SAN 5, a client contacts the server node to obtain metadata and locks. Metadata supplies the client with information about a file, such as its attributes and location on storage devices. Locks supply the client with privileges it needs to open a file and read or write data. The server node performs a look-up of metadata information for the requested file within the metadata space 25 of the SAN 5. The server node communicates granted lock information and file metadata to the requesting client node, including the location of all data blocks making up the file. Once the client node holds a distributed lock and knows the data block location(s), the client node can access the data for the file directly from a shared storage device attached to the SAN 5."

As such, Guthridge et al. discloses that a problem exists in such a configuration at col. 1, lines 36-45: "In distributed system with shared persistent storage, such as the configuration shown in FIG. 1, modifications to file metadata are coordinated between a client node and a server node using distributed locking techniques. However, problems arise when a client node holding a lock stops communicating with other nodes or when the server node is subject to a failure. The communication failure may be due to the client node experiencing a failure, or a network failure causing the client node holding the lock to be subject to a network partition."

Guthridge et al. teaches a solution at col. 3, lines 9-36: "In a distributed file system, locks are requested by client nodes and granted by a server node (**Note, this is completely different from the present invention**). All locks have a lock version number that is provided to the client node together with the grant of the lock. In addition, a server node instance is identified by a persistently stored epoch number, wherein the epoch number is incremented each time the server node begins managing the storage containing the data to be locked, such as when the server node is restarted

(**Note, this is completely different from the present invention**). When a client node is granted a lock from the server node, the client node establishes a lease with each server node serving a file system in use by the client node. A unique client node identifier is assigned to the client node when the lease is established. A granted lock is valid as long as a client node maintains a lease from the data server node that has granted the lock (**Note, this is completely different from the present invention**). If a client node fails to renew a lease for a granted lock, the lease will expire. The client node must obtain a new lease with a new client node identifier if they need the lock. If a client node should request a lock that is indicated as owned by a second client node in the system, but the second client node has failed to maintain the lock lease, the requesting client node may recover the lock from the second client node. This lock is known as a stolen lock. When a lock is stolen, the server node increments the lock version number on disk (**Note, this is completely different from the present invention**). Alternatively, if a lock lease has expired and has not been stolen, then a client node may try to reassert the lock. Accordingly, locks may be acquired or reasserted within various parameters while maintaining compatibility within the modes of the granted locks."

More specifically, Guthridge et al. teaches at col. 3, lines 44-46: "A distributed lock is obtained by a client node from a server node serving the file system which contains the file system object metadata. (**Note, this is completely different from the present invention**)" Guthridge et al. teaches at col. 3, lines 56-59: "When a server node is in a start-up mode of operation, it is assigned a file system to manage. For each file system that the server node is assigned, it must proceed through a routine to open the file system. (**Note, this is completely different from the present invention**)" Guthridge et al. teaches at col. 4, lines 13-15: "FIG. 3 is a flow chart 60 illustrating the process of a client acquiring a new lock from the server node. (**Note, this is completely different from the present invention**)" Guthridge et al. teaches at col. 4, lines 25-28: "For each object in the filesystem on which a client obtains a lock, the server node creates an in-memory data structure, hereinafter referred to as the lock manager object structure. (**Note, this is completely different from the present invention**)"

As such, Guthridge et al. discloses locks being requested by client nodes and granted by a server node (col. 3, lines 9-10). In contrast, the invention described in this patent application does not involve such a client-server relationship.

As to claim 1: The Examiner asserts:" **the first computing entity attempting to obtain a lease for itself on the first data entity by performing the reading the owner field** as a query is conducted to determine if a lock manager data structure exists (Guthridge, column 4, lines 29-31, Fig. 3A, element 70)." Applicants submit that the Examiner is wrong because Guthridge et al. states at col. 4, lines 25-32: "For each object in the filesystem on which a client obtains a lock, the **server node creates an in-memory data structure, hereinafter referred to as the lock manager object structure**. Following the step of reading the epoch number for the current file system, a query is conducted to determine if the lock manager data structure for the identified object exists in the lock manager cache on **the server node's volatile memory 70**. (Emphasis added)" As the Examiner can appreciate from this, Guthridge et al. teaches a method which is completely different from the present invention of claim 1 for at least two (2) reasons. **First**, Guthridge et al. teaches a server maintaining a lock manager data structure **whereas** claim 1 requires an owner field included in the file system. **Second**, Guthridge et al. teaches a server accessing lock information as an intermediary between a first computing entry and a file system **whereas** claim 1 requires that the first computing entry directly accesses the file system.

Applicants wish to point out that, because of the above identified differences between claim 1 and Guthridge et al., unpredictable advantages are obtained from one or more embodiments of claim 1 that are **not** obtainable from embodiments of the disclosure of Guthridge et al. In particular, embodiments of claim 1 enable multiple interacting hosts to get a lock by simply interacting, for example and without limitation, with a SCSI disk. Despite the fact that a SCSI disk system has a very limited and standardized interface, embodiments of claim 1 are able to use this very limited interface to achieve distributed locking. Guthridge et al. requires multiple computing entities to interact with the file system through an intermediary, a server, via IP networking. One advantage of providing locking information on the SCSI disk and providing direct access thereto is fewer configurations. Specifically, participating computing entities, for example, hosts, need only be configured to see the SCSI disk to be able to participate in the distributed file system. In prior art systems, such as those taught by Guthridge et al., client nodes, for example, hosts, need to be configured to see the SCSI disk, **AND** they must be configured (e.g. IP addresses) so they can talk to the locking server(s). In some computing systems, all hosts form a cluster to serve each other, and hence, must all be configured to know the IP addresses of all other hosts. Thus, adding or removing

a host can require much configuration. This is in contrast to systems embodied in accordance with claim 1 wherein no configuration to access the file system is required other than configuration to provide visibility to the shared disk. In addition, in embodiments of claim 1, availability of the file system (or lock) is exclusively dependent on the availability of the shared disk. However, Guthridge et al. teaches a file system (or lock) whose availability is a composite function of the availability of a connection between clients and a server and the availability of the shared disk. As such, embodiments of claim 1 are more reliable that embodiments of the system disclosed in Guthridge et al.

As to claim 1, the Examiner further asserts: "**if the owner field indicates that the first data entity is not currently leased, the first computing entity writing to the owner field to indicate an assumption of a lease of the first data entity** as if the lock data structure does not exist, a new lock manager data structure for the identified object is created (Guthridge, column 4, lines 33-35, Fig. 3A, element 76), client node identifier associated with the lease (Guthridge, column 6, lines 57-58), and **writing to the time field to indicate when the lease expires** as return file system epoch number of the lock (Guthridge, column 4, lines 58-59)." As the Examiner can appreciate from this, Guthridge et al. teaches a method which is completely different from the present invention of claim 1 for at least two (2) more reasons. **First**, Guthridge et al. teaches at col. 4, lines 33-35 that "Following the step of reading the epoch number for the current file system, a query is conducted to determine if the lock manager data structure for the identified object exists in the lock manager cache on the **server node's volatile memory 70** (Emphasis added)" **whereas** claim 1 requires writing to the owner field included **in the file system**. **Second**, Guthridge et al. teaches at col. 4, lines 58-59 the server node transmitting a file system epoch number of a lock to a client node **whereas** claim 1 requires writing to the time field included **in the file system**.

In Guthridge, at col. 3, lines 12-16: "In addition, a server node instance is identified by a persistently stored epoch number, wherein the epoch number is incremented each time the server node begins managing the storage containing the data to be locked, such as when the server node is restarted." As such, the epoch number of Guthridge et al. is **not** a time field of the present invention, and the two parameters are used in different ways.

Similarly, all assertions of the Examiner regarding reading or writing the time field and writing the owner field are different from Guthridge et al. for the same reasons discussed above.

Applicants agree that Taylor discloses at col. 2, lines 4-40 a lease data structure with a lease start time, lease length and determining that the lease time has expired. However, Applicants submit that even if one of ordinary skill in the art were to combine Guthridge and Taylor in the manner suggested by the Examiner, that person would not arrive at the invention of claim 1 or any claim dependent therefrom because that combination would still be missing the portions of the present invention wherein the owner field and the time field are included in the file system, and that the computing entities access the file system without intermediary computing entities as taught by Guthridge et al. and Taylor. As a result, a system provided by the Examiner's combination would not provide the present invention, and as such, nothing in Guthridge or Taylor that would make the invention of claim 1 obvious.

In light of the above, Applicants respectfully submit that claim 1 is patentable over Guthridge et al. in view of Taylor.

As to claim 2: Applicants submit that claim 2 depends from claim 1, and as such, is deemed to be patentable over Guthridge et al. in view of Taylor for the same reasons set forth above with respect to claim 1.

As to claim 3: Applicants submit that claim 3 depends from claims 1 and 2, and as such, is deemed to be patentable over Guthridge et al. in view of Taylor for the same reasons set forth above with respect to claims 1 and 2. In addition, neither Guthridge et al. nor Taylor provide any teaching or suggestion of any kind wherein "the first data entity includes metadata and the owner field is located in this metadata." Thus, even if Guthridge et al. and Taylor were combined in accordance with the Examiner's assertion, such a combination would not include this requirement of claim 3. Hence, nothing in Guthridge or Taylor would make the invention of claim 3 obvious.

As to claim 5: Applicants submit that claim 5 depends from claim 1, and as such, is deemed to be patentable over Guthridge et al. in view of Taylor for the same reasons set forth above with respect to claim 1. In addition, neither Guthridge et al. nor Taylor provide any teaching or suggestion of any kind wherein "the first computing entity writing a data value to the owner field [included in the first data entity] that uniquely identifies the first computing entity." Thus, even if Guthridge et al. and Taylor were combined in accordance with the Examiner's assertion, such a combination would not include this requirement of claim 5. Hence, nothing in Guthridge or Taylor would make the invention of claim 5 obvious.

As to claim 6: Applicants submit that claim 6 depends from claims 1 and 5, and as such, is deemed to be patentable over Guthridge et al. in view of Taylor for the same reasons set forth above with respect to claims 1 and 5. In addition, neither Guthridge et al. nor Taylor provide any teaching or suggestion of any kind wherein "the data value that uniquely identifies the first computing entity is determined autonomously by the first computing entity." Thus, even if Guthridge et al. and Taylor were combined in accordance with the Examiner's assertion, such a combination would not include this requirement of claim 6. Hence, nothing in Guthridge or Taylor would make the invention of claim 6 obvious.

As to claim 7: Applicants submit that claim 7 depends from claims 1 and 5, and as such, is deemed to be patentable over Guthridge et al. in view of Taylor for the same reasons set forth above with respect to claims 1 and 5. In addition, neither Guthridge et al. nor Taylor provide any teaching or suggestion of any kind wherein "wherein the owner field indicates that the first data entity is not currently leased when the owner field contains a value of zero." Thus, even if Guthridge et al. and Taylor were combined in accordance with the Examiner's assertion, such a combination would not include this requirement of claim 7. Hence, nothing in Guthridge or Taylor would make the invention of claim 7 obvious.

As to claim 9: Applicants submit that claim 9 depends from claim 1, and as such, is deemed to be patentable over Guthridge et al. in view of Taylor for the same reasons set forth above with respect to claim 1. In addition, neither Guthridge et al. nor Taylor provide any teaching or suggestion of any kind wherein "the first computing entity determines whether a prior lease has expired by reading a first value from the time field [in the file system], delaying for a predetermined lease period and reading a second value from the time field, wherein the first computing entity determines that the prior lease has expired if the second value is the same as the first value, and the first computing entity determines that the prior lease has not expired if the second value is different from the first value." Thus, even if Guthridge et al. and Taylor were combined in accordance with the Examiner's assertion, such a combination would not include this requirement of claim 9. Hence, nothing in Guthridge or Taylor would make the invention of claim 9 obvious.

As to claim 11: Applicants submit that claim 11 depends from claim 1, and is deemed to be patentable over Guthridge et al. in view of Taylor for the same reasons set forth above with respect to

claim 1. In addition, neither Guthridge et al. nor Taylor provide any teaching or suggestion of any kind "wherein, if the first computing entity concludes that the first data entity is currently unavailable, the first computing entity further writes an entry to a queue owner field in a queue in the file system to indicate an interest in accessing the first data entity. (Emphasis added)" Thus, even if Guthridge et al. and Taylor were combined in accordance with the Examiner's assertion, such a combination would not include this requirement of claim 11. Hence, nothing in Guthridge or Taylor would make the invention of claim 11 obvious.

As to claim 12: Applicants submit that claim 12 depends from claims 1 and 11, and is deemed to be patentable over Guthridge et al. in view of Taylor for the same reasons set forth above with respect to claims 1 and 11. In addition, neither Guthridge et al. nor Taylor provide any teaching or suggestion of any kind wherein "the first computing entity also writes to a queue time field in the queue in the file system to indicate a period of time for which the entry to the queue owner field is valid. (Emphasis added)" Thus, even if Guthridge et al. and Taylor were combined in accordance with the Examiner's assertion, such a combination would not include this requirement of claim 12. Hence, nothing in Guthridge or Taylor would make the invention of claim 12 obvious.

As to claim 14: Applicants submit that claim 14 depends from claim 1, and is deemed to be patentable over Guthridge et al. in view of Taylor for the same reasons set forth above with respect to claim 1. In addition, neither Guthridge et al. nor Taylor provide any teaching or suggestion of any kind "wherein, if a lease is obtained, the first computing entity also sets a renewal timer and, after the renewal timer expires, the first computing entity renews the lease by writing a new value to the time field [in the file system]." Thus, even if Guthridge et al. and Taylor were combined in accordance with the Examiner's assertion, such a combination would not include this requirement of claim 14. Hence, nothing in Guthridge or Taylor would make the invention of claim 14 obvious.

As to claim 30: As was set forth above with respect to claim 1, Guthridge et al. teaches a method which is completely different from the present invention of claim 30 for at least two (2) reasons. **First**, Guthridge et al. teaches a server maintaining a lock manager data structure **whereas** claim 30 requires an owner field included in the file system. **Second**, Guthridge et al. teaches a server accessing lock information as an intermediary between a first computing entry and a file system **whereas** claim 30 requires that the first computing entry directly accesses the file system.

In addition, as was set forth above with respect to claim 1, Applicants wish to point out that, because of these differences between claim 30 and Guthridge et al., similar unpredicted advantages (to those obtained from one or more embodiments of claim 1) are obtained from one or more embodiments of claim 30 that are likewise **not** obtainable from embodiments of the disclosure of Guthridge et al. for the same reasons.

As to claim 30, the Examiner further asserts: "**the first computing entity reading the owner field and determining whether the first data entity is in use by a computing entity** as a query is conducted to determine if a lock manager data structure exists (column 4, lines 29-31, Fig. 3A, element 70)" and "**if the first data entity is not in use by a computing entity, the first computing entity writing to the owner field to take control of a lock on the first data entity** as if the lock data structure does not exist, a new lock manager data structure for the identified object is created (column 4, lines 33-35, Fig. 3A, element 76), client node identifier associated with the lease (column 6, lines 57-58)." As the Examiner can appreciate from this, Guthridge et al. teaches a method which is completely different from the present invention of claim 30 for at least two (2) reasons. **First**, Guthridge et al. teaches a server maintaining a lock manager data structure **whereas** claim 30 requires an owner field included in the file system. **Second**, Guthridge et al. teaches a server accessing lock information as an intermediary between a first computing entry and a file system **whereas** claim 1 requires that the first computing entry directly accesses the file system.

As set forth above with respect to claim 1, Applicants submit that even if one of ordinary skill in the art were to combine Guthridge and Taylor in the manner suggested by the Examiner, that person would not arrive at the invention of claim 30 or any claim dependent therefrom because that combination would still be missing the portions of the present invention wherein the owner field and the time field are included in the file system, and that the computing entities access the file system without intermediary computing entities as taught by Guthridge et al. and Taylor. Thus, even if Guthridge et al. and Taylor were combined in accordance with the Examiner's assertion, such a combination would not include this requirement of claim 30. Hence, nothing in Guthridge or Taylor would make the invention of claim 30 obvious.

As to claim 31: Applicants submit that claim 31 depends from claim 30, and is deemed to be patentable over Guthridge et al. in view of Taylor for the same reasons set forth above with respect to

claim 30. In addition, neither Guthridge et al. nor Taylor provide any teaching or suggestion of any kind wherein "if the first data entity is in use by a computing entity, the first computing entity reading a time field in the file system to determine whether a lease on the data entity has expired and, if the lease has expired, the first computing entity writing to the owner field to break the existing lease and to indicate an assumption of a new lease of the first data entity. (Emphasis added)" Thus, even if Guthridge et al. and Taylor were combined in accordance with the Examiner's assertion, such a combination would not include this requirement of claim 31. Hence, nothing in Guthridge or Taylor would make the invention of claim 31 obvious.

As to claim 32: Applicants submit that claim 31 depends from claims 30 and 31, and is deemed to be patentable over Guthridge et al. in view of Taylor for the same reasons set forth above with respect to claims 30 and 31. In addition, neither Guthridge et al. nor Taylor provide any teaching or suggestion of any kind wherein "the first computing entity determines whether the lease has expired by reading a first value from the time field [in the file system], delaying for a predetermined lease period and reading a second value from the time field, wherein the first computing entity determines that the lease has expired if the second value is the same as the first value, and the first computing entity determines that the lease has not expired if the second value is different from the first value." Thus, even if Guthridge et al. and Taylor were combined in accordance with the Examiner's assertion, such a combination would not include this requirement of claim 32. Hence, nothing in Guthridge or Taylor would make the invention of claim 32 obvious.

As to claim 33: Applicants submit that claim 33 depends from claim 30, and is deemed to be patentable over Guthridge et al. in view of Taylor for the same reasons set forth above with respect to claim 30. In addition, neither Guthridge et al. nor Taylor provide any teaching or suggestion of any kind wherein "if the first data entity is not in use by a computing entity, in addition to writing to the owner field to take control of the lock on the first data entity, the first computing entity writing to a time field in the file system to indicate when a lease of the first data entity expires. (Emphasis added)" Thus, even if Guthridge et al. and Taylor were combined in accordance with the Examiner's assertion, such a combination would not include this requirement of claim 33. Hence, nothing in Guthridge or Taylor would make the invention of claim 33 obvious.

As to claim 34: Applicants submit that claim 34 depends from claim 30, and is deemed to be patentable over Guthridge et al. in view of Taylor for the same reasons set forth above with respect to claim 30.

As to claim 35: Applicants submit that claim 35 depends from claims 30 and 34, and is deemed to be patentable over Guthridge et al. in view of Taylor for the same reasons set forth above with respect to claims 30 and 34. In addition, neither Guthridge et al. nor Taylor provide any teaching or suggestion of any kind wherein "the first data entity includes metadata and the owner field is located in this metadata." Thus, even if Guthridge et al. and Taylor were combined in accordance with the Examiner's assertion, such a combination would not include this requirement of claim 35. Hence, nothing in Guthridge or Taylor would make the invention of claim 35 obvious.

As to claim 38: Applicants submit that claim 38 depends from claim 30, and is deemed to be patentable over Guthridge et al. in view of Taylor for the same reasons set forth above with respect to claim 30. In addition, neither Guthridge et al. nor Taylor provide any teaching or suggestion of any kind wherein "the first computing entity autonomously determines a data value that uniquely identifies the first computing entity and the first computing entity assumes a lock on the first data entity by writing the unique data value into the owner field [included in the file system]." Thus, even if Guthridge et al. and Taylor were combined in accordance with the Examiner's assertion, such a combination would not include this requirement of claim 38. Hence, nothing in Guthridge or Taylor would make the invention of claim 38 obvious.

As to claim 39: Applicants submit that claim 39 depends from claim 30, and is deemed to be patentable over Guthridge et al. in view of Taylor for the same reasons set forth above with respect to claim 30. In addition, neither Guthridge et al. nor Taylor provide any teaching or suggestion of any kind wherein "if control of the lock is not obtained, in addition to writing an entry to a queue owner field in the file system to indicate an interest in accessing the first data entity, the first computing entity writing to a queue time field in the file system to indicate a period of time for which the entry to the queue owner field is valid. (Emphasis added)" Thus, even if Guthridge et al. and Taylor were combined in accordance with the Examiner's assertion, such a combination would not include this requirement of claim 39. Hence, nothing in Guthridge or Taylor would make the invention of claim 39 obvious.

In light of the above, Applicants submit that claims 1-3, 5-7, 9, 11-12, 14, 30-35 and 38-39 are patentable over Guthridge et al. in view of Taylor. As such, Applicants request that the Examiner withdraw this rejection.

Claims 4, 10 and 36 are rejected under 35 U.S.C. 103(a). In particular, the Examiner stated:

> Claims 4, 10 and 36 are rejected under 35 U.S.C. 103(a) as being unpatentable over Guthridge and Taylor as applied to claims 1 and 30 and further in view of Shaughnessy (US 5,692,178) System and Methods for Improved File Management in a Multi-User Environment, granted November 25,1997, hereinafter Shaughnessy.
>
> In regard to claim 4, Guthridge and Taylor disclose different locks, such as session, data and range locks (Guthridge, column 3, lines 46-47), however Guthridge and Taylor do not specifically disclose **wherein the first data entity is a directory**. On the other hand, Shaughnessy discloses a plurality of locks types including a directory lock, (Shaughnessy, column 3, line 48).
>
> It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to modify the teaching of Guthridge to include the directory lock of Shaughnessy where **the first data entity is a directory** because a plurality of lock types are included for maximizing concurrent access while minimizing corruption and data loss (Shaughnessy, column 3, line 50-51). It is also noted that Guthridge, Taylor and Shaughnessy are from file management, and more specifically access management (Guthridge, Abstract: lines 1-2, locks in a distributed file system, Taylor, Abstract: line 2-3, locking mechanism to control access to a shared resource, Shaughnessy, Abstract: line 7, controlling concurrent access).
>
> In regard to claim 10, Guthridge and Taylor do not specifically disclose the method of **wherein the steps of reading the owner field and reading the time field are both performed in a single read operation.** On other hand, Shaughnessy discloses a lock file read in a single I/O operation (Shaughnessy, column 19, lines 15-16).
>
> It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to modify the teaching of Guthridge and Taylor to include the single I/O operation of Shaughnessy for **the steps of reading the owner field and reading the time field are both performed in a single read operation** because this avoids multiple reads, thus avoiding a performance penalty (Shaughnessy, column 19, lines 24-26).
>
> In regard to claim 36, Guthridge and Taylor disclose different locks, such as session, data and range locks (Guthridge, column 3, lines 46-47), however Guthridge and Taylor do not specifically disclose the method **wherein the first data entity is a directory.** On the other hand, Shaughnessy discloses a plurality of locks types including a directory lock, (Shaughnessy, column 3, line 48).
>
> It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to modify the teaching of Guthridge and Taylor to include the directory lock of Shaughnessy where **the first data entity is a directory** because a plurality of lock types are included for maximizing concurrent access while minimizing corruption and data loss (Shaughnessy, column 3, line 50-51).

Applicants respectfully traverse the Examiner's rejection.

Shaughnessy discloses a system for accessing a shared storage device that suffers from the same issues as Guthridge et al., namely, Shaughnessy teaches inserting an intermediary system between a client node and the shared storage system.

As to claim 4: Claim 4 depends from claim 1. As set forth above, none of Guthridge et al., Taylor, or Shaughnessy teach, hint or suggest the requirement of claim 1 that : "the first computing entity attempting to obtain a lease for itself on the first data entity by reading an owner field included in the file system that can be used to determine whether the first data entity is leased by a computing entity and a time field included in the file system that can be used to determine whether a lease for the first data entity has expired." Thus, even if Guthridge et al. and Taylor and Shaughnessy were combined in accordance with the Examiner's assertion, such a combination would not include these requirements of claim 4. Hence, nothing in Guthridge or Taylor or Shaughnessy would make the invention of claim 4 obvious.

As to claim 10: Claim 10 depends from claim 1, and requires "wherein the steps of reading the owner field and reading the time field are both performed in a single read operation." As set forth above, none of Guthridge et al., Taylor, or Shaughnessy teach, hint or suggest the requirement of claim 1 that : "the first computing entity attempting to obtain a lease for itself on the first data entity by reading an owner field included in the file system that can be used to determine whether the first data entity is leased by a computing entity and a time field included in the file system that can be used to determine whether a lease for the first data entity has expired"; and none of Guthridge et al., Taylor, or Shaughnessy teach, hint or suggest the requirement of claim 10 that "wherein the steps of reading the owner field and reading the time field are both performed in a single read operation." Thus, even if Guthridge et al. and Taylor and Shaughnessy were combined in accordance with the Examiner's assertion, such a combination would not include these requirements of claim 10. Hence, nothing in Guthridge or Taylor or Shaughnessy would make the invention of claim 10 obvious.

As to claim 36: Claim 36 depends from claim 30. As set forth above, none of Guthridge et al., Taylor, or Shaughnessy teach, hint or suggest the requirement of claim 30 that: "the first computing entity reading an owner field included in the file system-that can be used to determine whether the first data entity is in use by a computing entity and determining whether the first data entity is in use by a computing entity." Thus, even if Guthridge et al. and Taylor and Shaughnessy

were combined in accordance with the Examiner's assertion, such a combination would not include these requirements of claim 36. Hence, nothing in Guthridge or Taylor or Shaughnessy would make the invention of claim 36 obvious.

In light of the above, Applicants submit that claims 4, 10 and 36 are patentable over Guthridge et al. in view of Taylor and further in view of Shaughnessy. As such, Applicants request that the Examiner withdraw this rejection.

Claims 8, 13 and 37 are rejected under 35 U.S.C. 103(a). In particular, the Examiner stated:

> Claims 8, 13 and 37 are rejected under 35 U.S.C. 103(a) as being unpatentable over Guthridge and Taylor as applied to claims 1 and 30 and further in view of Stakutis et al. (US 6,658,417 B1) Term-Based Methods and Apparatus for Access to Files on Shared Storage Devices, granted December 2, 2003, hereinafter Stakutis.
>
> In regard to claim 8, Guthridge and Taylor disclose the method **wherein the first computing entity a lease expires a predetermined period of time after the lease begins** as locks have a lease for a limited time period (Guthridge, Abstract, lines 2-3), and however Guthridge does not specifically disclose **wherein the step of writing to the time field to indicate when the lease expires comprises the first computing entity writing a current time value to the time field**; although Guthridge does a lease for a limited time period (Guthridge, Abstract, lines 2-3). On the other hand, Stakutis discloses a lease duration referenced to the time of the request (Stakutis, column 10, lines 9-10).
>
> It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to modify the teaching of Guthridge to include the request time of Stakutis for **writing a current time value to the time field** because once a lease is granted, clients may let the lease expire rather then closing the session (Stakutis, column 10, lines 41-43). It is also noted that Guthridge, Taylor and Stakutis are from file management, and more specifically access management (Guthridge, Abstract: lines 1-2, locks in a distributed file system, Taylor, Abstract: line 2-3, locking mechanism to control access to a shared resource, Stakutis, Abstract: lines 15-16, access to file on the storage device by generating a "lease").
>
> In regard to claim 13, Guthridge and Taylor do not specifically disclose the method further comprising **the first computing entity reserving a disk on which the owner field and the time field are located to ensure exclusive access to the disk for the reading and writing of the owner field and the time field.** On the other hand, Stakutis discloses dedicated storage devices.
>
> It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to modify the teaching of Guthridge and Taylor to include the dedicated storage devices of Stakutis for **reserving a disk on which the owner field and the time field are located to ensure exclusive access to the disk for the reading and writing of the owner field and the time field** because this allows the client nodes to access the file system without extraneous network communications (Stakutis, column 4, lines 27-29).
>
> In regard to claim 37, Guthridge and Taylor do not specifically disclose the method further comprising **the first computing entity reserving a disk on which the owner field is located to ensure exclusive access to the disk for the reading and writing of the owner field.** On the other hand, Stakutis discloses dedicated storage devices.

> It would have been obvious to one of ordinary skill in the art at the time of applicant's invention to modify the teaching of Guthridge and Taylor to include the dedicated storage devices of Stakutis for **reserving a disk on which the owner field and the time field are located to ensure exclusive access to the disk for the reading and writing of the owner field and the time field** because this allows the client nodes to access the file system without extraneous network communications (Stakutis, column 4, lines 27-29).

Applicants respectfully traverse the Examiner's rejection.

Stakutis et al. discloses a system for accessing a shared storage device that suffers from the same issues as Guthridge et al., namely, Stakutis et al. teaches inserting an intermediary system between a client node and the shared storage system. Also, although Stakutis et al. teaches a system which is less intrusive in this respect than the system disclosed in Guthridge et al., the effect is the same, namely, the client node uses the intermediary (for example, a server) as a gateway to gain direct access to the shared storage. As noted above, this approach engenders substantial disadvantages when compared with one or more embodiments of the present invention in accordance with claims 8, 13 and 37.

As to claim 8: Claim 8 depends from claim 1, and further requires "wherein the first computing entity determines whether a prior lease has expired by reading a first value from the time field [included in the file system], delaying for a predetermined lease period and reading a second value from the time field, wherein the first computing entity determines that the prior lease has expired if the second value is the same as the first value, and the first computing entity determines that the prior lease has not expired if the second value is different from the first value." As set forth above, none of Guthridge et al., Taylor, or Stakutis et al. teach, hint or suggest the requirement of claim 1 that : "the first computing entity attempting to obtain a lease for itself on the first data entity by reading an owner field included in the file system that can be used to determine whether the first data entity is leased by a computing entity and a time field included in the file system that can be used to determine whether a lease for the first data entity has expired." Thus, none of Guthridge et al., Taylor, or Stakutis et al. teach, hint or suggest these requirements of claim 8. Thus, even if Guthridge et al. and Taylor and Stakutis et al. were combined in accordance with the Examiner's assertion, such a combination would not include these requirements of claim 8. Hence, nothing in Guthridge or Taylor or Stakutis et al. would make the invention of claim 8 obvious.

As to claim 13: Claim 13 depends from claim 1, and further requires "further comprising the first computing entity reserving a disk on which the owner field and the time field are located to

ensure exclusive access to the disk for the reading and writing of the owner field and the time field." As set forth above, none of Guthridge et al., Taylor, or Stakutis et al. teach, hint or suggest the requirement of claim 1 that : "the first computing entity attempting to obtain a lease for itself on the first data entity by reading an owner field included in the file system that can be used to determine whether the first data entity is leased by a computing entity and a time field included in the file system that can be used to determine whether a lease for the first data entity has expired." Thus, none of Guthridge et al., Taylor, or Stakutis et al. teach, hint or suggest these requirements of claim 13. Thus, even if Guthridge et al. and Taylor and Stakutis et al. were combined in accordance with the Examiner's assertion, such a combination would not include these requirements of claim 13. Hence, nothing in Guthridge or Taylor or Stakutis et al. would make the invention of claim 13 obvious.

As to claim 37: Claim 37 depends from claim 30, and further requires "further comprising the first computing entity reserving a disk on which the owner field and the time field are located to ensure exclusive access to the disk for the reading and writing of the owner field and the time field." As set forth above, none of Guthridge et al., Taylor, or Stakutis et al. teach, hint or suggest the requirement of claim 30 that : "the first computing entity reading an owner field included in the file system-that can be used to determine whether the first data entity is in use by a computing entity and determining whether the first data entity is in use by a computing entity." Thus, none of Guthridge et al., Taylor, or Stakutis et al. teach, hint or suggest these requirements of claims 37. Thus, even if Guthridge et al. and Taylor and Stakutis et al. were combined in accordance with the Examiner's assertion, such a combination would not include these requirements of claim 37.

In light of the above, Applicants submit that claims 8, 13 and 37 are patentable over Guthridge et al. in view of Taylor and further in view of Stakutis et al. As such, Applicants request that the Examiner withdraw this rejection.

Accordingly, Applicants submit that the present Application is in condition for allowance. Applicants therefore request reconsideration of the outstanding rejections and issue a Notice of Allowance. The Examiner is invited to contact the undersigned at 650-427-1052 to discuss any additional changes the Examiner may feel is necessary in light of this Amendment.

Date: March 30, 2009

Respectfully submitted,

/Michael B. Einschlag/

3401 Hillview Avenue
Palo Alto, California 94304
Phone: (650) 427-1052

Michael B. Einschlag
Reg. No. 29,301
Attorney for the Applicant